BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

Page 1, between lines 1 and 3, insert the following heading:

-- BACKGROUND OF THE INVENTION --.

Page 1, please amend the first paragraph as follows:

The invention is based on a method for compressing data[, as generically defined by the preamble to the main claim]. In the run-length encoding method (RLE method), a method for data compression is already known in which no data loss occurs. This method is used for example with so-called PCX data. The aim of the method is to combine repeating data elements and to store them along with the number of repetitions. If the data elements are bytes made up of eight bits, then the compression occurs in such a way that bytes up to a value of 63 which do not repeat are stored uncompressed; the two highest bits must have the value 0. Bytes with a value between 64 and 255, as well as bytes between zero and 63 which repeat, are stored encoded in a data element made up of two bytes. In the first byte in this instance, the highest two bits are set equal to one. The subsequent bits indicate the repetition factor for the second byte. In this method, it is disadvantageous that for individual bytes with a value between 64 and 255, the storage space requirement is doubled. In an unfavorable

case, compression using the RLE method can result in the required storage space being greater after compression than before compression. Also known for a data compression is the Lempel-Ziv algorithm, in which a data sequence to be compressed is tested for repetitions of partial sequences, repeating elements are stored in a code table, and the partial sequences are replaced by an assigned code symbol. A table of an unknown size must be set up for the decompression.

Page 1, between lines 20 and 25, amend the heading "Advantages of the Invention" to -- SUMMARY OF THE INVENTION --.

Amend the paragraph bridging pages 1 and 2 as follows:

The method according to the invention[, with the features of the main claim], has the advantage over the prior art that the information as to whether a data element is stored in a compressed or uncompressed form is stored in another additional data element. This makes it possible to use the compression method according to the invention for all data elements, independent of their value, since no additional compression information must be contained in the individual data elements themselves. It is also advantageous to embody the compression as dependent upon preceding

and subsequent data elements, since this permits a particularly effective compression.

Page 2, amend the first full paragraph on page 2 as follows:

[Advantageous modifications and improvements of the method disclosed in the main claim our possible by means of the steps taken in the dependent claims.] It is particularly advantageous that multiple identical data elements that repeat in sequence with one another are counted and stored with a repetition factor in the manner according to the invention. The repetition factor can then assume a maximal value of the data element since the information as to whether the data element is stored in a compressed or uncompressed form is stored in a different data element. It is also advantageous to store the compressed or uncompressed data elements in a storage region provided for this and to store the data element, which contains the information as to which data elements have been stored in a compressed or uncompressed form, in a different storage region since this storage permits an effective access to the data elements. With sequences of data elements which exceed a predeterminable size, it is also advantageous to execute a storage in different data packets. As a result, even graphics which require a great deal of storage space can be broken

down into data packets whose size, for example, is optimally adapted to the sector size of a data storage medium or the data packet size is adapted for transmission via an interface or via the Internet, e.g. via e-mail.

Page 4, line 11, change the heading "Drawings" to -- BRIEF DESCRIPTION OF THE DRAWINGS --.

Page 4, line 30, change the heading "Description of the Exemplary Embodiment" to -- DESCRIPTION OF THE PREFERRED EMBODIMENTS --.

Amended page 1 first paragraph:

The invention is based on a method for compressing data. In the run-length encoding method (RLE method), a method for data compression is already known in which no data loss occurs. This method is used for example with so-called PCX data. The aim of the method is to combine repeating data elements and to store them along with the number of repetitions. If the data elements are bytes made up of eight bits, then the compression occurs in such a way that bytes up to a value of 63 which do not repeat are stored uncompressed; the two highest bits must have the value 0. Bytes with a value between 64 and 255, as well as bytes between zero and 63 which repeat, are stored encoded in a data element made up of two bytes. In the first byte in this instance, the highest two bits are set equal to one. The subsequent bits indicate the repetition factor for the second byte. In this method, it is disadvantageous that for individual bytes with a value between 64 and 255, the storage space requirement is doubled. In an unfavorable case, compression using the RLE method can result in the required storage space being greater after compression than before compression. Also known for a data compression is the Lempel-Ziv algorithm, in which a data sequence to be compressed is tested for repetitions of partial sequences, repeating elements are stored in a code

table, and the partial sequences are replaced by an assigned code symbol. A table of an unknown size must be set up for the decompression.

Amended paragraph bridging pages 1 and 2 as follows:

The method according to the invention, has the advantage over the prior art that the information as to whether a data element is stored in a compressed or uncompressed form is stored in another additional data element. This makes it possible to use the compression method according to the invention for all data elements, independent of their value, since no additional compression information must be contained in the individual data elements themselves. It is also advantageous to embody the compression as dependent upon preceding and subsequent data elements, since this permits a particularly effective compression.

Amended page 2, first full paragraph:

It is particularly advantageous that multiple identical data elements that repeat in sequence with one another are counted and stored with a repetition factor in the manner according to the invention. The repetition factor can then assume a maximal value of the data element since

the information as to whether the data element is stored in a compressed or uncompressed form is stored in a different data element. It is also advantageous to store the compressed or uncompressed data elements in a storage region provided for this and to store the data element, which contains the information as to which data elements have been stored in a compressed or uncompressed form, in a different storage region since this storage permits an effective access to the data elements. With sequences of data elements which exceed a predeterminable size, it is also advantageous to execute a storage in different data packets. As a result, even graphics which require a great deal of storage space can be broken down into data packets whose size, for example, is optimally adapted to the sector size of a data storage medium or the data packet size is adapted for transmission via an interface or via the Internet, e.g. via e-mail.

CLAIMS

new claims:

15. A method of compressing a sequence of data elements [which are image data], comprising the steps of storing data elements in a compressed or uncompressed form depending on a correlation with the data elements preceding or following the data elements in a sequence; providing at least one additional data element in which is stored an information as to whether the stored data elements are stored in a compressed or incompressed form; storing in a first region the data element in which is stored information as to which data elements are stored in a compressed or uncompressed form; and storing in a second storage region the compressed or uncompressed data elements.

16. A method as defined in claim 1; and further comprising processing the sequence of data elements in a predeterminable order, in which successive elements are checked as to whether they are identical.

17. A method as defined in claim 15; and further comprising occurring a compression of the data element in such a way that data element

which occur several times in succession are counted and are stored with a repetition factor.

18. A method as defined in claim 15; and further comprising storing the sequence, in the sequences of data element which exceed a predeterminable number of data elements, in a number of data packets wherein each data packet is comprised of at least two data elements.

19. A method for decompressing a sequence of data elements which are image data, from a data packet, comprising the steps of comprising a data packet of data elements in a first region and a second region of the data packet; generating a sequence of data elements as a function of the data elements stored in the first region, from the data elements stored in the second region, with or without decompression; and processing the data elements and base elements in a predetermined sequence; respectively associating each base element of the data elements stored in the first region with two data elements stored in the second region; if a base element has a first value, not occurring a compression of the data element; if the base element has a second value not occurring a decompression.

20. A method as defined in claim 19; and further comprising, depending on the data elements present in the second region of the data packet and the first base element of a predetermined partial sequence to be processed according to a predetermined order which is an empty partial sequence, adding data elements; continuing generating the partial sequence for each additional base element to be processed as a function of the data element present in the second region of the data packet until a termination criterion is fulfilled.

21. A method as defined in claim 20; and further comprising, for the case in which no decompression occurs, adding data element to the partial sequence unchanged.

22. A method as defined in claim 20; and further comprising occurring a decompression in such a way that a first, predetermined data element associated with the base element is established as a repetition factor for a second, predetermined data element associated with the base element; and adding the second data element to the partial sequence in accordance with a repetition factor.

23. A method as defined in claim 20; and further comprising executing a decompression on a data sequence comprised of a number of concatenated or successive data packets.

24. A method as defined in claim 20; and further comprising connecting a device to a calculating unit and a display device; depending on information transmitted by the calculating unit, decompressing at least partially compressed sequences of data element

25. A method as defined in claim 24; and further comprising transmitting the consequently generated image data to the display device.

26. A method as defined in claim 25, wherein said transmitting includes transmitting via an image memory.

27. A method as defined in claim 24; and further comprising operating the device for compressing part of a freely programmable combination instrument.

Claims

1. A method for compressing a sequence of data elements, in particular image data, wherein the data elements are stored in a compressed or uncompressed form depending on the correlation with the data elements preceding or following the data element in the sequence, wherein at least one additional data element is provided in which is stored the information as to whether the stored data elements are stored in a compressed or uncompressed form.

2. The method according to claim 1, characterized in that the sequence of data elements is processed in a predeterminable order, in which successive elements are checked as to whether they are identical.

3. The method according to [one of the preceding claims] claim 1, characterized in that a compression of the data elements occurs in such a way that data elements which occur several times in succession are counted and are stored with a repetition factor.

4. The method according to [one of the preceding claims] claim 1, characterized in that the at least partially compressed sequence of data is stored in at least one memory (34).

5. The method according to claim 4, characterized in that the data element in which is stored information as to which data elements are stored in a compressed or uncompressed form is stored in a first storage region (11, 110) and that the compressed or uncompressed data elements are stored in a second storage region (12, 120).

6. The method according to [one of the preceding claims] claim 1, characterized in that in sequences of data elements which exceed a predeterminable number of data elements, the sequence is stored in a number of data packets (10, 100), wherein each data packet is comprised of at least two data elements.

7. A method for decompressing a sequence of data elements, in particular image data, from a data packet (10, 100), which is comprised of data elements in a first region (11, 110) and a second region (12, 120) of the data packet, wherein the sequence of data elements is generated as a function of the data elements stored in the first region, from the data elements stored in the second region, with or without a decompression.

8. The decompression method according to claim 7, characterized in that the data elements in the first region are comprised of base elements (15) and the data elements and base elements are processed in a predetermined sequence in that each base element of the data elements stored in the first region is respectively associated with two data elements (16, 17, 18) stored in the second region; if a base element has a first value, a decompression of the data elements does not occur, and if the base element has a second value, a decompression does occur.

9. The decompression method according to [one of claims 7 and 8] claim 7, characterized in that depending on the data elements present in the second region of the data packet (10, 100) and the first base element of a predetermined partial sequence to be processed according to the predetermined order, in particular an empty partial sequence, data elements are added and in that the partial sequence thus generated is continued with each additional base element to be processed, as a function of the data elements present in the second region of the data packet, until an termination criterion is fulfilled.

10. The decompression method according to claim 9, characterized in that for the case in which no decompression occurs, data elements are added to the partial sequence unchanged.

11. The decompression method according to [one of claims 9 and 10] claim 9, characterized in that a decompression occurs in such a way that a first, predetermined data element associated with the base element is established as a repetition factor for a second, predetermined data element associated with the base element and the second data element is added to the partial sequence in accordance with the repetition factor.

12. The decompression method according to [one claims 7 – 11] claim 7, characterized in that a decompression is executed on a data sequence comprised of a number of concatenated or successive data packets.

13. The decompression method according to [one claims 7 – 11] claim 7, characterized in that the device is connected to a calculating unit (31) and a display device (33), and that depending on information transmitted by the calculating unit (31), at least partially compressed sequences of data elements are decompressed and the consequently generated image data can be transmitted to the display device (33), preferably via an image memory (32).

14. The decompression method according to claim 13, characterized in that the device is for decompressing part of a freely programmable combination instrument.

In the claims:

Amend the claims as follows:

15. A method of compressing a sequence of data elements [which are image data], comprising the steps of storing data elements in a compressed or uncompressed form depending on a correlation with the data elements preceding or following the data elements in a sequence; providing at least one additional data element in which is stored an information as to whether the stored data elements are stored in a compressed or incompressed form; storing in a first region the data element in which is stored information as to which data elements are stored in a compressed or uncompressed form; and storing in a second storage region the compressed or uncompressed data elements.

19. A method for decompressing a sequence of data elements [which are image data], from a data packet, comprising the steps of comprising a data packet of data elements in a first region and a second region of the data packet; generating a sequence of data elements as a function of the data elements stored in the first region, from the data elements stored in the second region, with or without decompression; and

processing the data elements and base elements in a predetermined sequence; respectively associating each base element of the data elements stored in the first region with two data elements stored in the second region; if a base element has a first value, not occurring a compression of the data element; if the base element has a second value not occurring a decompression.

Amended claims:

15. A method of compressing a sequence of data elements, comprising the steps of storing data elements in a compressed or uncompressed form depending on a correlation with the data elements preceding or following the data elements in a sequence; providing at least one additional data element in which is stored an information as to whether the stored data elements are stored in a compressed or incompressed form; storing in a first region the data element in which is stored information as to which data elements are stored in a compressed or uncompressed form; and storing in a second storage region the compressed or uncompressed data elements.

19. A method for decompressing a sequence of data elements, from a data packet, comprising the steps of comprising a data packet of data elements in a first region and a second region of the data packet; generating a sequence of data elements as a function of the data elements stored in the first region, from the data elements stored in the second region, with or without decompression; and processing the data elements and base elements in a predetermined sequence; respectively associating each base element of the data elements stored in the first region

with two data elements stored in the second region; if a base element has a first value, not occurring a compression of the data element; if the base element has a second value not occurring a decompression.

Respectfully submitted,

Michael J. Striker
Reg. No. 27233
103 East Neck Road
Huntington, NY 11743
(631)549-4700